

EEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGGGGGG
EEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGGGGGG
EEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGGGGGG
EEE XXX XXX CCC HHH HHH NNN NNN NNN GGG
EEE XXX XXX CCC HHH HHH NNN NNN NNN GGG
EEE XXX XXX CCC HHH HHH NNN NNN NNN GGG
EEE XXX XXX CCC HHH HHH NNNNNNN NNN GGG
EEE XXX XXX CCC HHH HHH NNNNNNN NNN GGG
EEE XXX XXX CCC HHH HHH NNNNNNN NNN GGG
EEEEEEEEEEEE XXX XXX CCC HHHHHHHHHHHHHHHHH NNN NNN NNN GGG
EEEEEEEEEEEE XXX XXX CCC HHHHHHHHHHHHHHHHH NNN NNN NNN GGG
EEEEEEEEEEEE XXX XXX CCC HHHHHHHHHHHHHHHHH NNN NNN NNN GGG
EEE XXX XXX CCC HHH HHH NNN NNNNNNN GGG GGGGGGGGGGG
EEE XXX XXX CCC HHH HHH NNN NNNNNNN GGG GGGGGGGGGGG
EEE XXX XXX CCC HHH HHH NNN NNNNNNN GGG GGGGGGGGGGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG GGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG GGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG GGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG GGG
EEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGGGG
EEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGGGG
EEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGGGG

FILE ID**EXCLIB

J 15

0001 0
0002 0 MODULE exch\$library (IDENT = 'V04-000') = %TITLE 'Facility-wide library module'
0003 0 BEGIN
0004 0
0005 0 *****
0006 0 *
0007 0 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0008 0 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0009 0 * ALL RIGHTS RESERVED.
0010 0 *
0011 0 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0012 0 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0013 0 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0014 0 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0015 0 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0016 0 * TRANSFERRED.
0017 0 *
0018 0 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0019 0 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0020 0 * CORPORATION.
0021 0 *
0022 0 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0023 0 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0024 0 *
0025 0 *
0026 0 *****
0027 0
0028 0 ++
0029 0 FACILITY: EXCHANGE - Foreign volume interchange facility
0030 0
0031 0 ABSTRACT: BLISS Library for EXCHANGE facility
0032 0
0033 0 ENVIRONMENT: VAX/VMS User mode
0034 0
0035 0 AUTHOR: CW Hobbs , CREATION DATE: 1-July-1982
0036 0
0037 0 MODIFIED BY:
0038 0
0039 0 V03-002 CWH3002 CW Hobbs 12-Apr-1984
0040 0 Add NOREMOTE, NOTSAMEDEV and RT11_DIRSIZE message codes.
0041 0
0042 0
0043 0 --

L 15
16-Sep-1984 00:35:17
15-Sep-1984 22:44:13

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[EXCHNG.SRC]EXCLIB.B32;1 Page 2 (2)

```
0044 0 ! Include files:  
0045 0  
0046 0  
0047 0 ! LIBRARY files:  
0048 0  
0049 0 LIBRARY  
0050 0 'SYSSLIBRARY:LIB' ! VMS operating system library  
0051 0 :  
0052 0  
0053 0 ! REQUIRE files:  
0054 0  
0055 0 REQUIRE  
0056 0 'LIBS:EXCDEFS' ! include the SDL definitions  
0057 0 :  
1687 0  
1688 0 ! Macros:  
1689 0  
1690 0  
1691 0 ! Declare some macros as shorthand for the psect names  
1692 0  
1693 0 MACRO  
1694 0 $global_rw = PSECT GLOBAL = exch$rw_global (ADDRESSING_MODE (LONG_RELATIVE)); GLOBAL %  
1695 0 :
```

```
1696 0 ! Declare some common data structure initialization macros
1697 0
1698 0 MACRO
1699 0     ! Define shorthand for a single initialized dynamic string desc
1700 0
1701 0     $dyn_str_desc           ! Static declaration
1702 0     =
1703 0     BLOCK [dsc$k_d_bln,BYTE]
1704 0     PRESET ([dsc$b_class] = dsc$k_class_d,
1705 0             [dsc$b_dtype] = dsc$k_dtype_t,
1706 0             [dsc$w_length] = 0,
1707 0             [dsc$a_pointer] = 0)
1708 0     %,
1709 0
1710 0     $dyn_str_desc_init (desci)          ! Run-time initialization
1711 0     =
1712 0     BEGIN
1713 0     BIND
1714 0         desc = (desci) : VECTOR [2, LONG],
1715 0         tmpl = exch$gg_dyn_str_template : VECTOR [2, LONG];
1716 0         desc [0] = .tmpl [0];
1717 0         desc [1] = .tmpl [1];
1718 0     END
1719 0     %,
1720 0
1721 0     ! Define macro for a single initialized static string desc.
1722 0
1723 0     $stat_str_desc (L, A)           ! Static declaration
1724 0     =
1725 0     BLOCK [dsc$k_s_bln,BYTE]
1726 0     PRESET( [dsc$b_class] = dsc$k_class_s,
1727 0             [dsc$b_dtype] = dsc$k_dtype_t,
1728 0             [dsc$w_length] = (L),
1729 0             [dsc$a_pointer] = (A) )
1730 0     %,
1731 0
1732 0     $stat_str_desc_init (desci, L, A)    ! Run-time initialization
1733 0     =
1734 0     BEGIN
1735 0     BIND
1736 0         desc = (desci) : BLOCK [, BYTE];
1737 0         desc [dsc$b_class] = dsc$k_class_s;
1738 0         desc [dsc$b_dtype] = dsc$k_dtype_t;
1739 0         desc [dsc$w_length] = (L);
1740 0         desc [dsc$a_pointer] = (A);
1741 0     END
1742 0     %,
1743 0
1744 0     $str_desc_set (desci, L, A)      ! Copy new length and pointer fields (both static and dynamic)
1745 0     =
1746 0     BEGIN
1747 0     BIND
1748 0         desc = (desci) : BLOCK [, BYTE];
1749 0         desc [dsc$w_length] = (L);
1750 0         desc [dsc$a_pointer] = (A);
1751 0     END
1752 0     %,
```

```
1753 0
1754 0
1755 0
1756 0
M 1757 0
M 1758 0
1759 0
1760 0
1761 0
1762 0
1763 0
M 1764 0
M 1765 0
1766 0
1767 0
1768 0
1769 0
1770 0
M 1771 0
M 1772 0
1773 0
1774 0
1775 0
1776 0
1777 0
1778 0
1779 0
1780 0
1781 0
1782 0
1783 0
1784 0
1785 0
1786 0
M 1787 0
M 1788 0
M 1789 0
M 1790 0
M 1791 0
M 1792 0
M 1793 0
M 1794 0
M 1795 0
M 1796 0
M 1797 0
M 1798 0
M 1799 0
1800 0
1801 0
1802 0
1803 0
1804 0
1805 0
1806 0
M 1807 0
M 1808 0
M 1809 0

        ! And shorthand for just a descriptor declaration
        $desc_block
                =
                BLOCK [dsc$k_s_bln, BYTE]
                %,
        ! Short form for byte vector reference
        $ref_bvector
                =
                REF $bvector
                %,
        ! Short form for byte block reference
        $ref_bblock
                =
                REF $bblock
                %;

STRUCTURE
        $bvector [1; N] =
                [
                [N]
                ($bvector+I)<0,8,0>;

!+
! SIGNAL_STOP a condition assuming no return. LIB$exch_signal_STOP is not
! supposed to return, but BLISS doesn't know this, so we block further
! flow here. This will generate better code for us.
!-
MACRO
        Sexch_signal_stop []
                =
                BEGIN
                LINKAGE
                LNK = CALL : PRESERVE (0,1,2,3,4,5,6,7,8,9,10,11);
                EXTERNAL ROUTINE
                LIB$STOP : ADDRESSING_MODE (GENERAL) LNK NOVALUE;
                BUILTIN
                R0;
                LIB$STOP (%REMAINING);
                RETURN (.R0);
                END
                %;

!+
! SIGNAL a condition and return.
!-
MACRO
        Sexch_signal_return (code)
                =
                BEGIN
                LOCAL
```

```
; M 1810 0
; M 1811 0
; M 1812 0
; M 1813 0
; M 1814 0
; M 1815 0
; M 1816 0
; M 1817 0
; M 1818 0
; M 1819 0
; M 1820 0
; M 1821 0
; M 1822 0
; M 1823 0
; M 1824 0
; M 1825 0
; M 1826 0
; M 1827 0
; M 1828 0
; M 1829 0

        temp;
        temp = (code);      ! Need to avoid multiple calls, etc
        SIGNAL (.temp       ! Perform the actual signal of the error
                %IF %LENGTH GTR 1 %THEN ,%REMAINING %FI);
        RETURN .temp

        END
        %

;+ SIGNAL a condition and continue.
;-
MACRO $exch_signal (code)
        =
        SIGNAL ( (code)      ! Perform the actual signal of the error
                %IF %LENGTH GTR 1 %THEN ,%REMAINING %FI)
        %;
```

```
1830 0      !+ Initialize a control block type and size fields. We do not depend on them being in the standard positions
1831 0      !_-
1832 0      MACRO
1833 0          $block_init (addr, prefix)
1834 0              =
1835 0              BEGIN
1836 0              BIND
1837 0                  addr2 = (addr) : BLOCK [, BYTE];
1838 0                  addr2 [%NAME (prefix,'$w_size')] = %NAME ('exchblk$S_',prefix);
1839 0                  addr2 [%NAME (prefix,'$b_type')] = %NAME ('exchblk$K_',prefix);
1840 0              END
1841 0              %
1842 0
1843 0
1844 0      !+
1845 0      !_ Check a control block type and size fields. Note that we depend on them being in the standard positions
1846 0
1847 0      MACRO
1848 0          $block_check (level, addr, prefix, error_code)
1849 0              =
1850 0              %IF switch_variant GEQ (level)
1851 0                  %THEN
1852 0                      BEGIN
1853 0                          EXTERNAL ROUTINE
1854 0                              exch$util_block_check : jsb_r0r1r2 NOVALUE;
1855 0
1856 0                          exch$util_block_check ( (addr), (error_code),
1857 0                                  -(%NAME ('exchblk$S_',prefix) ^ 16 OR %NAME ('exchblk$K_',prefix)));
1858 0
1859 0                      END
1860 0
1861 0                  %FI
1862 0                  %
1863 0
1864 0      MACRO
1865 0          $block_check_if_nonzero (level, addr, prefix, error_code)
1866 0              =
1867 0              %IF switch_variant GEQ (level)
1868 0                  %THEN
1869 0                      BEGIN
1870 0                          BIND
1871 0                              addr2 = (addr) : BLOCK [, BYTE];
1872 0                              IF addr2 NEQ 0
1873 0                                  THEN $block_check ((level), (addr), (prefix), (error_code));
1874 0
1875 0                      END
1876 0
1877 0                  %FI
1878 0                  %
1879 0      !+
1880 0      !_ Check for a logic error. If the expression is not true, then we have a problem.
1881 0
1882 0      MACRO
1883 0          $logic_check (level, condition, error_code)
1884 0              =
1885 0
1886 0          ! See if a compile time check is possible
```

```
: M 1887 0
: M 1888 0
: M 1889 0
: M 1890 0
: M 1891 0
: M 1892 0
: M 1893 0
: M 1894 0
: M 1895 0
: M 1896 0
: M 1897 0
: M 1898 0
: M 1899 0
: M 1900 0
: M 1901 0
: M 1902 0
: M 1903 0
: M 1904 0
: M 1905 0
: M 1906 0
: M 1907 0
: M 1908 0
: M 1909 0
: M 1910 0
: M 1911 0
: M 1912 0
: M 1913 0
: M 1914 0
: M 1915 0
: M 1916 0
: M 1917 0
: M 1918 0
: M 1919 0
: M 1920 0
: M 1921 0
: M 1922 0
: M 1923 0

    XIF %CTCE ((condition))
    %THEN
        ! The condition is a compile-time expression. There is one special case, when the
        ! condition is the string "(false)". This is used as an unconditional logic abort.
        ! If we have "(false)", then do a naked SIGNAL_STOP
    XIF %IDENTICAL (condition, (false))
    %THEN
        SIGNAL_STOP (exch$_badlogic, 1, (error_code))
    ! The condition is a normal test. If it is true, print a message that the condition
    ! was verified during compilation. If false, generate a serious error.
    %ELSE
        %IF (condition)
        %THEN
            XPRINT ('assumption ',error_code,' verified during compilation')
        %ELSE
            XERROR ('assumption ',error_code,' is not true')
        %FI
    %FI
    ! The condition is not a compile-time constant. If the current variant calls for it,
    ! generate run-time code to test the assumption.
    %ELSE
        %IF switch_variant GEQ (level)
        %THEN
            BEGIN
                IF NOT (condition)
                THEN
                    SIGNAL_STOP (exch$_badlogic, 1, (error_code));
                END
        %FI
    %;

```

```

1924 0      +
1925 0      Most messages are defined as warnings so that we can signal without changing flow of execution. Several
1926 0      macros are defined in EXCLIB to change the severity code. These are $warning_stat, $success_stat,
1927 0      $error_stat, $info_stat and $severe_stat. A typical use would be:
1928 0
1929 0      status = lib$foo (bar);
1930 0      IF NOT .status
1931 0      THEN
1932 0          BEGIN
1933 0          $warning_stat (.status);      ! Convert unknown severity to warning
1934 0          SIGNAL (.status);
1935 0          RETURN .status;
1936 0          END;
1937 0
1938 0      These macros modify the status variable and return the value of the modified status.
1939 0
1940 0      For those situations where it is inappropriate to modify the code, forms of the macro are available which
1941 0      modify copies of the status code. These macro names have "_copy" appended to the modify form of the macro
1942 0      name. A couple of examples of their use are:
1943 0
1944 0      status = lib$foo (bar);           status = lib$foo (bar);
1945 0      IF NOT .status                IF NOT .status
1946 0      THEN
1947 0          BEGIN
1948 0          new_stat = $warning_status_copy (.status);    BEGIN
1949 0          SIGNAL (.new_stat);           SIGNAL ($error_status_copy (exch$_badfoo), 0, .status);
1950 0          RETURN (.status);           RETURN (exch$_badfoo);
1951 0          END;
1952 0
1953 0      Note that the "_copy" forms have value arguments, the regular forms have address arguments.
1954 0      -
1955 0
1956 0      | Convert status codes to specific status values.
1957 0
1958 0      MACRO
1959 0          $inhibit_msg (status)
M 1960 0          =
M 1961 0          BEGIN
M 1962 0          BIND
M 1963 0          status2 = status : BLOCK [4, BYTE];
M 1964 0          status2 [sts$v_inhib_msg] = 1;      ! Inhibit SEXIT signalling
M 1965 0          .status2                      ! Value of block is new code
M 1966 0          END
M 1967 0          %
M 1968 0
M 1969 0          $inhibit_msg_copy (status)
M 1970 0          =
M 1971 0          BEGIN
M 1972 0          LOCAL
M 1973 0          status2 : BLOCK [4, BYTE];
M 1974 0          status2 [0,0,32,0] = status;      ! Copy the whole code
M 1975 0          status2 [sts$v_inhib_msg] = 1;    ! Inhibit SEXIT signalling
M 1976 0          .status2                      ! Value of block is new code
M 1977 0          END
M 1978 0          %
M 1979 0
M 1980 0          $warning_stat (status)

```

```

M 1981 0
M 1982 0
M 1983 0
M 1984 0
M 1985 0
M 1986 0
M 1987 0
M 1988 0
M 1989 0
M 1990 0
M 1991 0
M 1992 0
M 1993 0
M 1994 0
M 1995 0
M 1996 0
M 1997 0
M 1998 0
M 1999 0
M 2000 0
M 2001 0
M 2002 0
M 2003 0
M 2004 0
M 2005 0
M 2006 0
M 2007 0
M 2008 0
M 2009 0
M 2010 0
M 2011 0
M 2012 0
M 2013 0
M 2014 0
M 2015 0
M 2016 0
M 2017 0
M 2018 0
M 2019 0
M 2020 0
M 2021 0
M 2022 0
M 2023 0
M 2024 0
M 2025 0
M 2026 0
M 2027 0
M 2028 0
M 2029 0
M 2030 0
M 2031 0
M 2032 0
M 2033 0
M 2034 0
M 2035 0
M 2036 0
M 2037 0

= BEGIN
BIND
    status2 = status : BLOCK [4, BYTE];
    status2 [sts$v_severity] = sts$k_warning; ! Force status to warning
    .status2 ! Value of block is new code
END
%,

$warning_stat_copy (status)
=
BEGIN
LOCAL
    status2 : BLOCK [4, BYTE];
    status2 [0,0,32,0] = status;
    status2 [sts$v_severity] = sts$k_warning; ! Copy the whole code
    ! Force status to warning
    ! Value of block is new code
.END
%,

$success_stat (status)
=
BEGIN
BIND
    status2 = status : BLOCK [4, BYTE];
    status2 [sts$v_severity] = sts$k_success; ! Force status to success
    .status2 ! Value of block is new code
END
%,

$success_stat_copy (status)
=
BEGIN
LOCAL
    status2 : BLOCK [4, BYTE];
    status2 [0,0,32,0] = status;
    status2 [sts$v_severity] = sts$k_success; ! Copy the whole code
    ! Force status to success
    ! Value of block is new code
.END
%,

$error_stat (status)
=
BEGIN
BIND
    status2 = status : BLOCK [4, BYTE];
    status2 [sts$v_severity] = sts$k_error; ! Force status to error
    .status2 ! Value of block is new code
END
%,

$error_stat_copy (status)
=
BEGIN
LOCAL
    status2 : BLOCK [4, BYTE];
    status2 [0,0,32,0] = status; ! Copy the whole code

```

```

: M 2038 0
: M 2039 0
: M 2040 0
: M 2041 0
: M 2042 0
: M 2043 0
: M 2044 0
: M 2045 0
: M 2046 0
: M 2047 0
: M 2048 0
: M 2049 0
: M 2050 0
: M 2051 0
: M 2052 0
: M 2053 0
: M 2054 0
: M 2055 0
: M 2056 0
: M 2057 0
: M 2058 0
: M 2059 0
: M 2060 0
: M 2061 0
: M 2062 0
: M 2063 0
: M 2064 0
: M 2065 0
: M 2066 0
: M 2067 0
: M 2068 0
: M 2069 0
: M 2070 0
: M 2071 0
: M 2072 0
: M 2073 0
: M 2074 0
: M 2075 0
: M 2076 0
: M 2077 0
: M 2078 0
: M 2079 0
: M 2080 0
: M 2081 0
: M 2082 0
: M 2083 0
: M 2084 0
: M 2085 0
: M 2086 0
: M 2087 0
: M 2088 0
: M 2089 0
: M 2090 0
: M 2091 0
: M 2092 0
: M 2093 0
: M 2094 0

        status2 [sts$v_severity] = sts$k_error; ! Force status to error
        .status2                                         ! Value of block is new code
        END
        %.

        $info_stat (status)
        =
        BEGIN
        BIND
            status2 = status : BLOCK [4, BYTE];
            status2 [sts$v_severity] = sts$k_info; ! Force status to info
            .status2                                         ! Value of block is new code
        END
        %.

        $info_stat_copy (status)
        =
        BEGIN
        LOCAL
            status2 : BLOCK [4, BYTE];
            status2 [0,0,32,0] = status;           ! Copy the whole code
            status2 [sts$v_severity] = sts$k_info; ! Force status to info
            .status2                                         ! Value of block is new code
        END
        %.

        $severe_stat (status)
        =
        BEGIN
        BIND
            status2 = status : BLOCK [4, BYTE];
            status2 [sts$v_severity] = sts$k_severe; ! Force status to severe
            .status2                                         ! Value of block is new code
        END
        %.

        $severe_stat_copy (status)
        =
        BEGIN
        LOCAL
            status2 : BLOCK [4, BYTE];
            status2 [0,0,32,0] = status;           ! Copy the whole code
            status2 [sts$v_severity] = sts$k_severe; ! Force status to severe
            .status2                                         ! Value of block is new code
        END
        %;

        ! Special debug and trace macros
        ! MACRO
        $dbgtrc_prefix (string)                  ! Declare a nested macro with the value of the string
        =
        MACRO $dbgtrc_prefix_string = string %QUOTE %
        %.

        $check_call (level, routine_addr)          ! Call the routine depending on variant level

```

H 16
16-Sep-1984 00:35:17
15-Sep-1984 22:44:15

VAX-11 Bliss-32 v4.0-742
_S255\$DUA28:[EXCHNG.SRC]EXCLIB.B32;1 Page 11 (5)

2095 0
2096 0
2097 0
2098 0
2099 0
2100 0
2101 0
2102 0
2103 0

=%IF switch_variant GEQ (level)
%THEN -
BEGIN
EXTERNAL ROUTINE routine_addr : ADDRESSING_MODE (GENERAL);
routine_addr (%REMAINING)
END;
ZF I
%;

```
2104 0 ! Message print routines
2105 0
2106 0 MACRO
2107 0     $print_lit (string)
2108 0         =
2109 0         lib$put_output (%ASCIID string)
2110 0         %,
2111 0
2112 0     $trace_print_lit (string)
2113 0         =
2114 0         %IF switch_trace
2115 0         %THEN
2116 0             lib$put_output (%ASCIID %STRING ($dbgtrc_prefix_string, string))
2117 0         %FI ! switch_trace
2118 0         %,
2119 0
2120 0     $debug_print_lit (string)
2121 0         =
2122 0         %IF switch_debug
2123 0         %THEN
2124 0             lib$put_output (%ASCIID %STRING ($dbgtrc_prefix_string, string))
2125 0         %FI ! switch_debug
2126 0         %,
2127 0
2128 0     $print_desc (desc)
2129 0         =
2130 0         lib$put_output (desc)
2131 0         %,
2132 0
2133 0     $trace_print_desc (desc)
2134 0         =
2135 0         %IF switch_trace
2136 0         %THEN
2137 0             BEGIN
2138 0                 EXTERNAL ROUTINE exch$util_fao_buffer;
2139 0                 lib$put_output (
2140 0                     exch$util_fao_buffer (%ASCIID %STRING ($dbgtrc_prefix_string, '!AS'), desc))
2141 0                 END
2142 0             %FI ! switch_trace
2143 0             %,
2144 0
2145 0     $debug_print_desc (desc)
2146 0         =
2147 0         %IF switch_debug
2148 0         %THEN
2149 0             BEGIN
2150 0                 EXTERNAL ROUTINE exch$util_fao_buffer;
2151 0                 lib$put_output (
2152 0                     exch$util_fao_buffer (%ASCIID %STRING ($dbgtrc_prefix_string, '!AS'), desc));
2153 0                 END
2154 0             %FI ! switch_debug
2155 0             %,
2156 0
2157 0     $print_fao (string)
2158 0         =
2159 0         BEGIN
2160 0             EXTERNAL ROUTINE exch$util_fao_buffer;
```

```
2161 0
2162 0
2163 0
2164 0
2165 0
2166 0
2167 0
2168 0
2169 0
2170 0
2171 0
2172 0
2173 0
2174 0
2175 0
2176 0
2177 0
2178 0
2179 0
2180 0
2181 0
2182 0
2183 0
2184 0
2185 0
2186 0
2187 0
2188 0
2189 0
2190 0
2191 0

lib$put_output (
    exch$util_fao_buffer (%ASCID string
    %IF %LENGTH GTR 1 %THEN ,%REMAINING %FI))
END
%,
Strace_print_fao (string)
=
%IF switch_trace
%THEN
    BEGIN
        EXTERNAL ROUTINE exch$util_fao_buffer;
        lib$put_output (
            exch$util_fao_buffer (%ASCID %STRING ($dbgtrc_prefix_string, string)
            %IF %LENGTH GTR 1 %THEN ,%REMAINING %FI))
    END
    %FI ! switch_trace
%,
Sdebug_print_fao (string)
=
%IF switch_debug
%THEN
    BEGIN
        EXTERNAL ROUTINE exch$util_fao_buffer;
        lib$put_output (
            exch$util_fao_buffer (%ASCID %STRING ($dbgtrc_prefix_string, string)
            %IF %LENGTH GTR 1 %THEN ,%REMAINING %FI))
    END
    %FI ! switch_debug
%;
```

```
2192 0      ! Macros to manipulate queues
2193 0
2194 0      MACRO
2195 0
2196 0      ! Initialize the header of a queue. This means make each of the 2 pointers in the header point to the header.
2197 0
2198 0      $queue_initialize (q_header)
2199 0      =
2200 0      BEGIN
2201 0
2202 0      BIND
2203 0      _qh_ = (q_header) : VECTOR [2, LONG];
2204 0
2205 0      _qh_[0] = _qh_;
2206 0      _qh_[1] = _qh_;
2207 0
2208 0      END
2209 0      %
2210 0
2211 0      ! Insert an element at the head of a queue.
2212 0
2213 0      $queue_insert_head (item, q_header)
2214 0      =
2215 0      BEGIN
2216 0
2217 0      BUILTIN
2218 0      INSQUE;
2219 0
2220 0      BIND
2221 0      _qh_ = (q_header) : VECTOR [2, LONG];
2222 0
2223 0      INSQUE ((item), _qh_[0])
2224 0
2225 0      END
2226 0      %
2227 0
2228 0      ! Insert an element at the tail of a queue.
2229 0
2230 0      $queue_insert_tail (item, q_header)
2231 0      =
2232 0      BEGIN
2233 0
2234 0      BUILTIN
2235 0      INSQUE;
2236 0
2237 0      BIND
2238 0      _qh_ = (q_header) : VECTOR [2, LONG];
2239 0
2240 0      INSQUE ((item), ._qh_[1])
2241 0
2242 0      END
2243 0      %
2244 0      ! Remove the indicated element from a queue. The first parameter is the address of the element. The second
2245 0      parameter is optional.
2246 0
2247 0      If supplied, it is the address of a longword in which to store the element removed from the queue or 0 if
2248 0      no element was present in the queue. The value of the expression is TRUE if a element was removed from the
```

```
2249 0
2250 0
2251 0
2252 0
2253 0
2254 0
2255 0
2256 0
2257 0
2258 0
2259 0
2260 0
2261 0
2262 0
2263 0
2264 0
2265 0
2266 0
2267 0
2268 0
2269 0
2270 0
2271 0
2272 0
2273 0
2274 0
2275 0
2276 0
2277 0
2278 0
2279 0
2280 0
2281 0
2282 0
2283 0
2284 0
2285 0
2286 0
2287 0
2288 0
2289 0
2290 0
2291 0
2292 0
2293 0
2294 0
2295 0
2296 0
2297 0
2298 0
2299 0
2300 0
2301 0
2302 0
2303 0
2304 0
2305 0

| If the second parameter is not supplied, the value of the expression is the address of the element removed
| from the queue or 0 if no element was present in the queue.
|- Queue_remove (q_element, element)
= BEGIN
  BIND _ghead_ = (q_element) : VECTOR [2, LONG];
  BUILTIN REMQUE;
  %IF (%NULL (element))
  %THEN LOCAL _T_ : REF VECTOR [2, LONG];
  %ELSE _T_ = (element) : REF VECTOR [2, LONG];
  %FI
  IF (REMQUE (_ghead_, _T_))
  THEN BEGIN
    ! queue was empty
    IF (%NULL (element))
    THEN 0
    ELSE (_T_ = 0; FALSE)
  END
  ELSE BEGIN
    IF (%NULL (element))
    THEN
      _T_
    ELSE true
  END
  END
%.

| Remove an element from the head of a queue. The first parameter is the address of the queue header. The
| second parameter is optional.
If supplied, it is the address of a longword in which to store the element removed from the queue or 0 if
no element was present in the queue. The value of the expression is TRUE if an element was removed from the
queue and FALSE otherwise.
If the second parameter is not supplied, the value of the expression is the address of the element removed
from the queue or 0 if no element was present in the queue.
|- Queue_remove_head (q_header, element)
```

```
2306 0
2307 0
2308 0
2309 0
2310 0
2311 0
2312 0
2313 0
2314 0
2315 0
2316 0
2317 0
2318 0
2319 0
2320 0
2321 0
2322 0
2323 0
2324 0
2325 0
2326 0
2327 0
2328 0
2329 0
2330 0
2331 0
2332 0
2333 0
2334 0
2335 0
2336 0
2337 0
2338 0
2339 0
2340 0
2341 0
2342 0
2343 0
2344 0
2345 0
2346 0
2347 0
2348 0
2349 0
2350 0
2351 0
2352 0
2353 0
2354 0
2355 0
2356 0
2357 0
2358 0
2359 0
2360 0
2361 0

= BEGIN
BIND
    _qh_ = (q_header) : VECTOR [2, LONG];
%IF (%NULL (element))
%THEN
    $queue_remove (.qh_[0])
%ELSE
    $queue_remove (.qh_[0], element)
%FI
END
%.

+ Remove an element from the tail of a queue. The first parameter is the address of the queue header. The second parameter is optional.
If supplied, it is the address of a longword in which to store the element removed from the queue or 0 if no element was present in the queue. The value of the expression is TRUE if an element was removed from the queue and FALSE otherwise.
If the second parameter is not supplied, the value of the expression is the address of the element removed from the queue or 0 if no element was present in the queue.

$queue_remove_tail (q_header, element)
=
BEGIN
BIND
    _qh_ = (q_header) : VECTOR [2, LONG];
%IF (%NULL (element))
%THEN
    $queue_remove (.qh_[1])
%ELSE
    $queue_remove (.qh_[1], element)
%FI
END
%.

! Test a queue for emptiness. TRUE if the queue is empty, FALSE if the queue is not empty
$queue_empty (q_header)
=
BEGIN
BIND
    _qh_ = (q_header) : VECTOR [2, LONG];
    _qh_ EQLA .qh_[0]
END
%;
```

```

2362 0
2363 0 | Literal definitions:
2364 0
2365 0
2366 0 | define literals for BLISS true and false values
2367 0
2368 0 LITERAL
2369 0   true = 1
2370 0   false = 0
2371 0   ;
2372 0
2373 0 | Define values of some ASCII characters
2374 0
2375 0 LITERAL
2376 0   NUL = 0
2377 0   LF = 10,
2378 0   VT = 11.
2379 0   FF = 12.
2380 0   CR = 13
2381 0   CTRLZ = 26,
2382 0   ESC = 27
2383 0   DEL = 127
2384 0   ;
2385 0
2386 0 | Define the Radix-50 equivalents for FILE.BAD
2387 0
2388 0 LITERAL
2389 0   R50_EMPTY = %RAD50_11 'EMPTY '.
2390 0   R50_FIL = %RAD50_11 'FIL'.
2391 0   R50_FILE = %X '1F4026F4',
2392 0   R50_BAD = %X '0CAC'.
2393 0   R50_SYS = %X '7ABB'
2394 0   ;
2395 0
2396 0
2397 0 | Linkage definitions:
2398 0
2399 0 LINKAGE
2400 0   jsb_r0r1 = JSB (REGISTER=0, REGISTER=1)
2401 0   : NOPRESERVE(0,1) NOTUSED(2,3,4,5,6,7,8,9,10,11),
2402 0   jsb_r0r1r2 = JSB (REGISTER=0, REGISTER=1, REGISTER=2)
2403 0   : NOPRESERVE(0,1,2) NOTUSED(3,4,5,6,7,8,9,10,11),
2404 0   jsb_r1 = JSB (REGISTER=1)
2405 0   : NOPRESERVE(0,1) NOTUSED(2,3,4,5,6,7,8,9,10,11),
2406 0   jsb_r1r2 = JSB (REGISTER=1, REGISTER=2)
2407 0   : NOPRESERVE(0,1,2) NOTUSED(3,4,5,6,7,8,9,10,11),
2408 0   jsb_r1r2r3 = JSB (REGISTER=1, REGISTER=2, REGISTER=3)
2409 0   : NOPRESERVE(0,1,2,3) NOTUSED(4,5,6,7,8,9,10,11),
2410 0   jsb_r2r3 = JSB (REGISTER=2, REGISTER=3)
2411 0   : NOPRESERVE(0,1,2,3) NOTUSED(4,5,6,7,8,9,10,11),
2412 0   jsb_r3r4 = JSB (REGISTER=3, REGISTER=4)
2413 0   : NOPRESERVE(0,1,2,3,4) NOTUSED(5,6,7,8,9,10,11),
2414 0   jsb_get = JSB (REGISTER=5, REGISTER=6, REGISTER=7)
2415 0   : NOPRESERVE(0,1,2,3,4,5,6,7) NOTUSED(8,9,10,11),
2416 0   jsb_put = JSB (REGISTER=9, REGISTER=10)
2417 0   : NOPRESERVE(0,1,2,3,4,5,6,7,8,9,10) NOTUSED(11)
2418 0

```

2419 0 ; Run-time library and other routines external to the facility
2420 0
2421 0 EXTERNAL ROUTINE
2422 0 cli\$acl_parse : ADDRESSING_MODE (GENERAL), | Command parsing routine
2423 0 cli\$dispatch : ADDRESSING_MODE (GENERAL), | Action routine dispatch
2424 0 cli\$get_value : ADDRESSING_MODE (GENERAL), | Entity value fetch
2425 0 cli\$present : ADDRESSING_MODE (GENERAL), | Entity presence boolean
2426 0 lib\$find_file : ADDRESSING_MODE (GENERAL), | Wildcard files-11 processing
2427 0 lib\$free_vm : ADDRESSING_MODE (GENERAL), | Releases memory
2428 0 lib\$get_input : ADDRESSING_MODE (GENERAL), | Get a line from SYSSINPUT
2429 0 lib\$get_vm : ADDRESSING_MODE (GENERAL), | Gets memory
2430 0 lib\$put_output: ADDRESSING_MODE (GENERAL), | Display a line on SYSSOUTPUT
2431 0 ots\$cvt_tf_ : ADDRESSING_MODE (GENERAL), | ASCII decimal to longword
2432 0 ots\$cvt_to_l : ADDRESSING_MODE (GENERAL), | ASCII octal to longword
2433 0 ots\$cvt_tz_l : ADDRESSING_MODE (GENERAL), | ASCII hexadecimal to longword
2434 0 str\$copy_dx : ADDRESSING_MODE (GENERAL), | Copy string of any class
2435 0 str\$freeT_dx : ADDRESSING_MODE (GENERAL) | Release dynamic string
2436 0 ;
2437 0
2438 0 ! Define the lengths of control blocks here - Many of these need to be adjusted by system block sizes, so it
2439 0 can't be completely done in the SDL definition.
2440 0
2441 0 LITERAL
2442 0
2443 0 ! An SEXCG is the global environment for the facility, the SDL block plus two RMS work areas
2444 0
2445 0 exchblk\$\$_excg = excg\$k_length + 2*(fab\$k_bln + rab\$k_bln + nam\$k_bln + (2*nam\$c_maxrss)),
2446 0
2447 0 ! An SRMSB describes an RMS file, the SDL block plus one RMS work area
2448 0
2449 0 exchblk\$\$_rmsb = rmsb\$k_length + fab\$k_bln + rab\$k_bln + nam\$k_bln + (2*nam\$c_maxrss),
2450 0
2451 0 ! A SVOLB contains the structures for a volume, the SDL block plus one RMS work area
2452 0
2453 0 exchblk\$\$_volb = volb\$k_length + fab\$k_bln + rab\$k_bln + nam\$k_bln + (2*nam\$c_maxrss),
2454 0
2455 0 ! The following don't need adjusting, but we want to keep all the EXCHBLK\$\$_ definitions in one place
2456 0
2457 0 exchblk\$\$_copy = copy\$k_length, | Size of the work area for the COPY command
2458 0 exchblk\$\$_dire = dire\$k_length, | Size of the work area for the DIRECTORY command
2459 0 exchblk\$\$_dos11 = dos11\$k_length, | Size of the DOS-11 specific extension to the volb
2460 0 exchblk\$\$_dos11ctx = dos11Tctx\$k_length, | Size of the DOS-11 file context block
2461 0 exchblk\$\$_filb = filb\$k_length, | A SFILB is a structure which describes an open file
2462 0 exchblk\$\$_init = init\$k_length, | Size of the work area for the INIT command
2463 0 exchblk\$\$_namb = namb\$k_length, | A \$NAMB is a structure which stores a fully parsed file name
2464 0 exchblk\$\$_moun = moun\$k_length, | Size of the work area for the MOUNT command
2465 0 exchblk\$\$_rt11 = rt11\$k_length, | Size of the RT-11 specific extension to the volb
2466 0 exchblk\$\$_rt11ctx = rt11Tctx\$k_length, | Size of the RT-11 file context block
2467 0 exchblk\$\$_rtnam = rtnam\$k_length | Size of the work area for the DIRECTORY command
2468 0 ;

```

2469 0 ; Message codes defined in SRC$:EXCMSG.MSG
2470 0
2471 0 EXTERNAL LITERAL
2472 0 exchs_accessfail, failed to access volume ($GETDVI service failure)
2473 0 exchs_badfilename, File name not valid for given volume
2474 0 exchs_badlogic, Adds error number to shared message
2475 0 exchs_badpad, Improper /RECORD FORMAT=PAD option
2476 0 exchs_binchksum, Bad formatted binary record
2477 0 exchs_binrecfmt, Bad formatted binary record
2478 0 exchs_blockcheck, Block check failed
2479 0 exchs_blockcheck0, Block check failed because block address is 0
2480 0 exchs_canceled, Command canceled
2481 0 exchs_closeerr, Error closing file
2482 0 exchs_closeforeign, Error closing foreign device
2483 0 exchs_copied, Log message for copy command
2484 0 exchs_copyboot, Log message for copy /boot command
2485 0 exchs_copnewname, File copied with new name
2486 0 exchs_createvirt, Error creating virtual volume
2487 0 exchs_deleted, Deleted copy of a file
2488 0 exchs_deleteprev, Deleted previous copy of a file
2489 0 exchs_devonly, Device spec only, other parts of file name ignored
2490 0 exchs_devnotsuit, Device is not suitable for EXCHANGE
2491 0 exchs_dire_error, Error writing directory
2492 0 exchs_dismounted, Device has been dismounted
2493 0 exchs_dos11_badlabel, Invalid label found on dos11 tape
2494 0 exchs_dos11_blocksize, Invalid block (>512 bytes) found on dos11 tape
2495 0 exchs_dos11_ierror, Error during I/O on dos11 tape
2496 0 exchs_dos11_position, Rewinding tape to find correct position
2497 0 exchs_filenotfound, Unable to locate file
2498 0 exchs_fil11_norec, No /RECORD for files-11
2499 0 exchs_ignore_dire, Ignoring directory specification
2500 0 exchs_ignore_vers, Ignoring file version number
2501 0 exchs_illmtcopy, Illegal magtape copy, input and output on same device
2502 0 exchs_initialized, Device has been initialized
2503 0 exchs_invrectfmt, Record format not valid for volume type
2504 0 exchs_invvolfmt, Volume format not valid for operation
2505 0 exchs_many_to_one, Multiple input files were given but only one output file
2506 0 exchs_mounted, Volume mounted (success)
2507 0 exchs_mounterror, Error performing VMS $mount service
2508 0 exchs_mountvir, Virtual volume mounted (success)
2509 0 exchs_noalloc, /ALLOCATE ignored on tape output
2510 0 exchs_nocarriage, /CARRIAGE ignored on output
2511 0 exchs_nocopbad, Couldn't create, .BAD file with wildcared names
2512 0 exchs_nocopbadel, Couldn't create, have to delete .BAD file
2513 0 exchs_nocopdup, Couldn't create, already created same name
2514 0 exchs_nocoplock, Couldn't create, volume is writelocked
2515 0 exchs_nocopnodef, Couldn't create, file of same name and /NODELETE given
2516 0 exchs_nocopprot, Couldn't create, file of same name protected against modification
2517 0 exchs_nocopsamdev, Illegal copy to same device
2518 0 exchs_nocopsysdel, Illegal copy of .SYS when existing .SYS present
2519 0 exchs_nocopyboot, Unable to copy boot info
2520 0 exchs_nodellock, File not deleted, volume locked
2521 0 exchs_nodevice, Device spec missing
2522 0 exchs_noremove, Device spec cannot have node field
2523 0 exchs_norendev, Illegal rename to different device
2524 0 exchs_norenexists, Not renamed, already exists
2525 0 exchs_norenlock, Files not renamed, volume locked

```

2526	0	exch\$_nosysact,	No action on .SYS files
2527	0	exch\$_notcopied,	File not copied
2528	0	exch\$_notcop_retry,	File not copied, will retry
2529	0	exch\$_notdeleted,	File not deleted
2530	0	exch\$_notimplement,	! Feature not yet implemented
2531	0	exch\$_notmounted,	Device is not mounted on EXCHANGE
2532	0	exch\$_notsamedev,	Input and output not same device (copy /boot)
2533	0	exch\$_notvallen,	/REC=LEN requires FIXED
2534	0	exch\$_novolumes,	No volumes are mounted
2535	0	exch\$_openforeign,	Open failed on a foreign volume
2536	0	exch\$_openvirtual,	Open failed on a virtual volume
2537	0	exch\$_opnotperdos,	Operation not permitted on DOS-11 volume
2538	0	exch\$_opnotperf11,	Operation not permitted on Files-11 volume
2539	0	exch\$_opnotperrt11,	Operation not permitted on RT-11 volume (not yet needed)
2540	0	exch\$_opnotperrtmt,	Operation not permitted on RT-11 magtape volume
2541	0	exch\$_parseerr,	Bad file parameter syntax
2542	0	exch\$_partcopied,	File partially copied
2543	0	exch\$_readcheck,	Error detected during read check
2544	0	exch\$_readcheckrec,	Error detected during read check was recovered
2545	0	exch\$_readerrrec,	Error detected during read was recovered
2546	0	exch\$_recover,	Directory recovery message
2547	0	exch\$_rectoobig,	Bad formatted binary record
2548	0	exch\$_renamed,	File renamed log message
2549	0	exch\$_rt11_baddirct,	RT-11 directory error
2550	0	exch\$_rt11_badfile,	Bad block file created
2551	0	exch\$_rt11_bigbadfile,	Bad block file contains some good blocks
2552	0	exch\$_rt11_dirsize,	Device size disagrees with directory size
2553	0	exch\$_rt11_errlock,	RT-11 directory error
2554	0	exch\$_rt11_extra,	Too many extra words
2555	0	exch\$_rt11_noend,	RT-11 directory error
2556	0	exch\$_rt11_overflow,	RT-11 directory error
2557	0	exch\$_rt11_stblock,	RT-11 directory error
2558	0	exch\$_rt11_toomanyblk,	RT-11 directory error
2559	0	exch\$_rt11_toomanyseg,	RT-11 directory error
2560	0	exch\$_rt11_unkent,	RT-11 directory error
2561	0	exch\$_rtoufeof,	End of file on output file
2562	0	exch\$_rtprotect,	File protected against modification
2563	0	exch\$_stmrecfmt,	Bad stream record format
2564	0	exch\$_stnotavail,	Start block not available
2565	0	exch\$_strtnomulti,	Can't say /START with multiple input files
2566	0	exch\$_toomanycol,	Too many columns requested
2567	0	exch\$_trace,	Header for a status trace
2568	0	exch\$_typed,	Log message for type command
2569	0	exch\$_virtnochange,	Cannot change size of virtual devices
2570	0	exch\$_vmsmount,	Volume has been mounted on VMS
2571	0	exch\$_volmount,	Volume is already mounted
2572	0	exch\$_volume_full,	Output volume is full
2573	0	exch\$_waiterr,	Error waiting for RMS operation
2574	0	exch\$_writecache,	Writing modified directory segments
2575	0	exch\$_writecheck,	Error detected during write check
2576	0	exch\$_writecheckrec,	Error detected during write check was recovered
2577	0	exch\$_writeerrrec,	Error detected during write was recovered
2578	0	exch\$_writelock	Volume is write-locked
2579	0	:	
2580	0		
2581	0	Shared message definitions	
2582	0	:	

```

: P 2583 0 $shr msgdef
: P 2584 0   `exch, 248, local,
: P 2585 0     ! (badlogic, warning),
: P 2586 0     (badvalue, warning),
: P 2587 0     (closeout, warning),
: P 2588 0     (configual, warning)
: P 2589 0     (insvirmem, warning),
: P 2590 0     (openin, warning),
: P 2591 0     (openout, warning),
: P 2592 0     (readerr, warning),
: P 2593 0     (writeerr, warning)
: 2594 0   );
: 2595 0
: P 2596 0 $shr msgdef
: P 2597 0   `msg, 3, local,
: P 2598 0     {syntax, severe}
: 2599 0   );
: 2600 0
: 2601 0   ! Other symbols which need explicit declarations
: 2602 0
: 2603 0 EXTERNAL LITERAL
: 2604 0   clis_comma,
: 2605 0   clis_concat,
: 2606 0   clis_locneg,
: 2607 0   clis_locpres,
: 2608 0   clis_nocomd,
: 2609 0   clis_negated,
: 2610 0   clis_present,
: 2611 0   clis_facility;
: 2612 0
: 2613 0   ! Storage external to all modules
: 2614 0
: 2615 0 EXTERNAL
: 2616 0   exch$cld_table : ADDRESSING_MODE (LONG_RELATIVE)           ! Command table for CLISDCL_PARSE
: 2617 0   ;
: 2618 0
: 2619 0   ! External data - defined in EXCHSMAIN module
: 2620 0
: 2621 0 EXTERNAL
: 2622 0   exch$qq_dyn_str_template : $desc block ADDRESSING_MODE (LONG_RELATIVE).      ! An initialized, null dynamic strin
: 2623 0   exch$sa_gbl : REF BLOCK [,BYTE] ADDRESSING_MODE (LONG_RELATIVE)          ! The pointer to the known world
: 2624 0   ;
: 2625 0
: 2626 0   ! END
: 2627 0   ! ELUDOM
: 2628 0
: 2629 0
: 2630 0
: 2631 0
: 2632 0
: 2633 0
: 2634 0
: 2635 0
: 2636 0
: 2637 0
: 2638 0
: 2639 0
: 2640 0
: 2641 0
: 2642 0
: 2643 0
: 2644 0
: 2645 0
: 2646 0
: 2647 0
: 2648 0
: 2649 0
: 2650 0
: 2651 0
: 2652 0
: 2653 0
: 2654 0
: 2655 0
: 2656 0
: 2657 0
: 2658 0
: 2659 0
: 2660 0
: 2661 0
: 2662 0
: 2663 0
: 2664 0
: 2665 0
: 2666 0
: 2667 0
: 2668 0
: 2669 0
: 2670 0
: 2671 0
: 2672 0
: 2673 0
: 2674 0
: 2675 0
: 2676 0
: 2677 0
: 2678 0
: 2679 0
: 2680 0
: 2681 0
: 2682 0
: 2683 0
: 2684 0
: 2685 0
: 2686 0
: 2687 0
: 2688 0
: 2689 0
: 2690 0
: 2691 0
: 2692 0
: 2693 0
: 2694 0
: 2695 0
: 2696 0
: 2697 0
: 2698 0
: 2699 0
: 2700 0
: 2701 0
: 2702 0
: 2703 0
: 2704 0
: 2705 0
: 2706 0
: 2707 0
: 2708 0
: 2709 0
: 2710 0
: 2711 0
: 2712 0
: 2713 0
: 2714 0
: 2715 0
: 2716 0
: 2717 0
: 2718 0
: 2719 0
: 2720 0
: 2721 0
: 2722 0
: 2723 0
: 2724 0
: 2725 0
: 2726 0
: 2727 0
: 2728 0
: 2729 0
: 2730 0
: 2731 0
: 2732 0
: 2733 0
: 2734 0
: 2735 0
: 2736 0
: 2737 0
: 2738 0
: 2739 0
: 2740 0
: 2741 0
: 2742 0
: 2743 0
: 2744 0
: 2745 0
: 2746 0
: 2747 0
: 2748 0
: 2749 0
: 2750 0
: 2751 0
: 2752 0
: 2753 0
: 2754 0
: 2755 0
: 2756 0
: 2757 0
: 2758 0
: 2759 0
: 2760 0
: 2761 0
: 2762 0
: 2763 0
: 2764 0
: 2765 0
: 2766 0
: 2767 0
: 2768 0
: 2769 0
: 2770 0
: 2771 0
: 2772 0
: 2773 0
: 2774 0
: 2775 0
: 2776 0
: 2777 0
: 2778 0
: 2779 0
: 2780 0
: 2781 0
: 2782 0
: 2783 0
: 2784 0
: 2785 0
: 2786 0
: 2787 0
: 2788 0
: 2789 0
: 2790 0
: 2791 0
: 2792 0
: 2793 0
: 2794 0
: 2795 0
: 2796 0
: 2797 0
: 2798 0
: 2799 0
: 2800 0
: 2801 0
: 2802 0
: 2803 0
: 2804 0
: 2805 0
: 2806 0
: 2807 0
: 2808 0
: 2809 0
: 2810 0
: 2811 0
: 2812 0
: 2813 0
: 2814 0
: 2815 0
: 2816 0
: 2817 0
: 2818 0
: 2819 0
: 2820 0
: 2821 0
: 2822 0
: 2823 0
: 2824 0
: 2825 0
: 2826 0
: 2827 0
: 2828 0
: 2829 0
: 2830 0
: 2831 0
: 2832 0
: 2833 0
: 2834 0
: 2835 0
: 2836 0
: 2837 0
: 2838 0
: 2839 0
: 2840 0
: 2841 0
: 2842 0
: 2843 0
: 2844 0
: 2845 0
: 2846 0
: 2847 0
: 2848 0
: 2849 0
: 2850 0
: 2851 0
: 2852 0
: 2853 0
: 2854 0
: 2855 0
: 2856 0
: 2857 0
: 2858 0
: 2859 0
: 2860 0
: 2861 0
: 2862 0
: 2863 0
: 2864 0
: 2865 0
: 2866 0
: 2867 0
: 2868 0
: 2869 0
: 2870 0
: 2871 0
: 2872 0
: 2873 0
: 2874 0
: 2875 0
: 2876 0
: 2877 0
: 2878 0
: 2879 0
: 2880 0
: 2881 0
: 2882 0
: 2883 0
: 2884 0
: 2885 0
: 2886 0
: 2887 0
: 2888 0
: 2889 0
: 2890 0
: 2891 0
: 2892 0
: 2893 0
: 2894 0
: 2895 0
: 2896 0
: 2897 0
: 2898 0
: 2899 0
: 2900 0
: 2901 0
: 2902 0
: 2903 0
: 2904 0
: 2905 0
: 2906 0
: 2907 0
: 2908 0
: 2909 0
: 2910 0
: 2911 0
: 2912 0
: 2913 0
: 2914 0
: 2915 0
: 2916 0
: 2917 0
: 2918 0
: 2919 0
: 2920 0
: 2921 0
: 2922 0
: 2923 0
: 2924 0
: 2925 0
: 2926 0
: 2927 0
: 2928 0
: 2929 0
: 2930 0
: 2931 0
: 2932 0
: 2933 0
: 2934 0
: 2935 0
: 2936 0
: 2937 0
: 2938 0
: 2939 0
: 2940 0
: 2941 0
: 2942 0
: 2943 0
: 2944 0
: 2945 0
: 2946 0
: 2947 0
: 2948 0
: 2949 0
: 2950 0
: 2951 0
: 2952 0
: 2953 0
: 2954 0
: 2955 0
: 2956 0
: 2957 0
: 2958 0
: 2959 0
: 2960 0
: 2961 0
: 2962 0
: 2963 0
: 2964 0
: 2965 0
: 2966 0
: 2967 0
: 2968 0
: 2969 0
: 2970 0
: 2971 0
: 2972 0
: 2973 0
: 2974 0
: 2975 0
: 2976 0
: 2977 0
: 2978 0
: 2979 0
: 2980 0
: 2981 0
: 2982 0
: 2983 0
: 2984 0
: 2985 0
: 2986 0
: 2987 0
: 2988 0
: 2989 0
: 2990 0
: 2991 0
: 2992 0
: 2993 0
: 2994 0
: 2995 0
: 2996 0
: 2997 0
: 2998 0
: 2999 0
: 3000 0
: 3001 0
: 3002 0
: 3003 0
: 3004 0
: 3005 0
: 3006 0
: 3007 0
: 3008 0
: 3009 0
: 3010 0
: 3011 0
: 3012 0
: 3013 0
: 3014 0
: 3015 0
: 3016 0
: 3017 0
: 3018 0
: 3019 0
: 3020 0
: 3021 0
: 3022 0
: 3023 0
: 3024 0
: 3025 0
: 3026 0
: 3027 0
: 3028 0
: 3029 0
: 3030 0
: 3031 0
: 3032 0
: 3033 0
: 3034 0
: 3035 0
: 3036 0
: 3037 0
: 3038 0
: 3039 0
: 3040 0
: 3041 0
: 3042 0
: 3043 0
: 3044 0
: 3045 0
: 3046 0
: 3047 0
: 3048 0
: 3049 0
: 3050 0
: 3051 0
: 3052 0
: 3053 0
: 3054 0
: 3055 0
: 3056 0
: 3057 0
: 3058 0
: 3059 0
: 3060 0
: 3061 0
: 3062 0
: 3063 0
: 3064 0
: 3065 0
: 3066 0
: 3067 0
: 3068 0
: 3069 0
: 3070 0
: 3071 0
: 3072 0
: 3073 0
: 3074 0
: 3075 0
: 3076 0
: 3077 0
: 3078 0
: 3079 0
: 3080 0
: 3081 0
: 3082 0
: 3083 0
: 3084 0
: 3085 0
: 3086 0
: 3087 0
: 3088 0
: 3089 0
: 3090 0
: 3091 0
: 3092 0
: 3093 0
: 3094 0
: 3095 0
: 3096 0
: 3097 0
: 3098 0
: 3099 0
: 3100 0
: 3101 0
: 3102 0
: 3103 0
: 3104 0
: 3105 0
: 3106 0
: 3107 0
: 3108 0
: 3109 0
: 3110 0
: 3111 0
: 3112 0
: 3113 0
: 3114 0
: 3115 0
: 3116 0
: 3117 0
: 3118 0
: 3119 0
: 3120 0
: 3121 0
: 3122 0
: 3123 0
: 3124 0
: 3125 0
: 3126 0
: 3127 0
: 3128 0
: 3129 0
: 3130 0
: 3131 0
: 3132 0
: 3133 0
: 3134 0
: 3135 0
: 3136 0
: 3137 0
: 3138 0
: 3139 0
: 3140 0
: 3141 0
: 3142 0
: 3143 0
: 3144 0
: 3145 0
: 3146 0
: 3147 0
: 3148 0
: 3149 0
: 3150 0
: 3151 0
: 3152 0
: 3153 0
: 3154 0
: 3155 0
: 3156 0
: 3157 0
: 3158 0
: 3159 0
: 3160 0
: 3161 0
: 3162 0
: 3163 0
: 3164 0
: 3165 0
: 3166 0
: 3167 0
: 3168 0
: 3169 0
: 3170 0
: 3171 0
: 3172 0
: 3173 0
: 3174 0
: 3175 0
: 3176 0
: 3177 0
: 3178 0
: 3179 0
: 3180 0
: 3181 0
: 3182 0
: 3183 0
: 3184 0
: 3185 0
: 3186 0
: 3187 0
: 3188 0
: 3189 0
: 3190 0
: 3191 0
: 3192 0
: 3193 0
: 3194 0
: 3195 0
: 3196 0
: 3197 0
: 3198 0
: 3199 0
: 3200 0
: 3201 0
: 3202 0
: 3203 0
: 3204 0
: 3205 0
: 3206 0
: 3207 0
: 3208 0
: 3209 0
: 3210 0
: 3211 0
: 3212 0
: 3213 0
: 3214 0
: 3215 0
: 3216 0
: 3217 0
: 3218 0
: 3219 0
: 3220 0
: 3221 0
: 3222 0
: 3223 0
: 3224 0
: 3225 0
: 3226 0
: 3227 0
: 3228 0
: 3229 0
: 3230 0
: 3231 0
: 3232 0
: 3233 0
: 3234 0
: 3235 0
: 3236 0
: 3237 0
: 3238 0
: 3239 0
: 3240 0
: 3241 0
: 3242 0
: 3243 0
: 3244 0
: 3245 0
: 3246 0
: 3247 0
: 3248 0
: 3249 0
: 3250 0
: 3251 0
: 3252 0
: 3253 0
: 3254 0
: 3255 0
: 3256 0
: 3257 0
: 3258 0
: 3259 0
: 3260 0
: 3261 0
: 3262 0
: 3263 0
: 3264 0
: 3265 0
: 3266 0
: 3267 0
: 3268 0
: 3269 0
: 3270 0
: 3271 0
: 3272 0
: 3273 0
: 3274 0
: 3275 0
: 3276 0
: 3277 0
: 3278 0
: 3279 0
: 3280 0
: 3281 0
: 3282 0
: 3283 0
: 3284 0
: 3285 0
: 3286 0
: 3287 0
: 3288 0
: 3289 0
: 3290 0
: 3291 0
: 3292 0
: 3293 0
: 3294 0
: 3295 0
: 3296 0
: 3297 0
: 3298 0
: 3299 0
: 3300 0
: 3301 0
: 3302 0
: 3303 0
: 3304 0
: 3305 0
: 3306 0
: 3307 0
: 3308 0
: 3309 0
: 3310 0
: 3311 0
: 3312 0
: 3313 0
: 3314 0
: 3315 0
: 3316 0
: 3317 0
: 3318 0
: 3319 0
: 3320 0
: 3321 0
: 3322 0
: 3323 0
: 3324 0
: 3325 0
: 3326 0
: 3327 0
: 3328 0
: 3329 0
: 3330 0
: 3331 0
: 3332 0
: 3333 0
: 3334 0
: 3335 0
: 3336 0
: 3337 0
: 3338 0
: 3339 0
: 3340 0
: 3341 0
: 3342 0
: 3343 0
: 3344 0
: 3345 0
: 3346 0
: 3347 0
: 3348 0
: 3349 0
: 3350 0
: 3351 0
: 3352 0
: 3353 0
: 3354 0
: 3355 0
: 3356 0
: 3357 0
: 3358 0
: 3359 0
: 3360 0
: 3361 0
: 3362 0
: 3363 0
: 3364 0
: 3365 0
: 3366 0
: 3367 0
: 3368 0
: 3369 0
: 3370 0
: 3371 0
: 3372 0
: 3373 0
: 3374 0
: 3375 0
: 3376 0
: 3377 0
: 3378 0
: 3379 0
: 3380 0
: 3381 0
: 3382 0
: 3383 0
: 3384 0
: 3385 0
: 3386 0
: 3387 0
: 3388 0
: 3389 0
: 3390 0
: 3391 0
: 3392 0
: 3393 0
: 3394 0
: 3395 0

```

COMMAND QUALIFIERS

: BLISS/LIBRARY=LIB\$:/LIST=LIS\$: SRC\$:EXCLIB

: Run Time: 00:21.2
: Elapsed Time: 01:09.7
: Lines/CPU Min: 7434
: Lexemes/CPU-Min: 40975
: Memory Used: 279 pages
: Library Precompilation Complete

0161 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

EXCFILE
LIS

EXCINIT
LIS

EXCLIB
LIS

EXCIO
LIS

0162 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

